

NPS55-87-015

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



AN OPTIMAL BRANCH-AND-BOUND PROCEDURE FOR THE  
CONSTRAINED PATH, MOVING TARGET SEARCH PROBLEM

JAMES N. EAGLE  
JAMES R. YEE

DECEMBER 1987

Approved for public release; distribution unlimited.

Prepared for:

Chief of Naval Operations  
Washington, DC 20350-2000

FedDocs  
D 208.14/2  
NPS-55-87-015

NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA

Rear Admiral R. C. Austin  
Superintendent

K. T. Marshall  
Acting Provost

The work reported herein was supported in part with funds provided from the Chief of Naval Operations.

Reproduction of all or part of this report is authorized.

This report was prepared by:

## REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>		1b. RESTRICTIVE MARKINGS	
SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited	
DECLASSIFICATION / DOWNGRADING SCHEDULE		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
PERFORMING ORGANIZATION REPORT NUMBER(S) 55-87-015		7a. NAME OF MONITORING ORGANIZATION	
NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b. OFFICE SYMBOL (If applicable) Code 55	
ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b. ADDRESS (City, State, and ZIP Code)	
NAME OF FUNDING / SPONSORING ORGANIZATION Chief of Naval Operations		8b. OFFICE SYMBOL (If applicable) OP-953	
ADDRESS (City, State, and ZIP Code) Washington, D. C. 20350-2000		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N41756D7WP78122	
TITLE (Include Security Classification) OPTIMAL BRANCH-AND-BOUND PROCEDURE FOR THE CONSTRAINED PATH, MOVING TARGET SEARCH PROBLEM		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
PERSONAL AUTHOR(S) Eagle, James N. and Yee, James R. (University of Southern California)		TASK NO.	WORK UNIT ACCESSION NO.
		14. DATE OF REPORT (Year, Month, Day) 1987 Dec	
TYPE OF REPORT Technical		13b. TIME COVERED FROM _____ TO _____	
SUPPLEMENTARY NOTATION		15. PAGE COUNT 8	
COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>A search is conducted for a target moving in discrete time among a finite number of cells according to a known Markov process. The searcher must choose one cell in which to search each time period. The set of cells available for search depends upon the cell chosen in the last time period. The problem is to find a search path, i.e., a sequence of search cells that maximizes the probability of detecting the target in a fixed number of time periods. Closely following earlier work by Theodor Stewart, a branch-and-bound procedure is developed which finds optimal search paths. This procedure is tested and appears to be more efficient than existing dynamic programming solution methods.</p>			
DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
NAME OF RESPONSIBLE INDIVIDUAL James N. Eagle		22b. TELEPHONE (Include Area Code) (408) 646-2654	
		22c. OFFICE SYMBOL Code 55Er	



# **An Optimal Branch-and-Bound Procedure for the Constrained Path, Moving Target Search Problem**

by

James N. Eagle  
Department of Operations Research  
Naval Postgraduate School  
Monterey, CA 93943

James R. Yee  
Department of Electrical Engineering-Systems  
University of Southern California  
Los Angeles, CA 90089

## **ABSTRACT**

A search is conducted for a target moving in discrete time among a finite number of cells according to a known Markov process. The searcher must choose one cell in which to search in each time period. The set of cells available for search depends upon the cell chosen in the last time period. The problem is to find a search path, i.e., a sequence of search cells, that maximizes the probability of detecting the target in a fixed number of time periods. Closely following earlier work by Theodor Stewart, a branch-and-bound procedure is developed which finds optimal search paths. This procedure is tested and appears to be more efficient than existing dynamic programming solution methods.





# An Optimal Branch-and-Bound Procedure for the Constrained Path, Moving Target Search Problem

by

James N. Eagle  
Department of Operations Research  
Naval Postgraduate School  
Monterey, CA 93943

James R. Yee  
Department of Electrical Engineering-Systems  
University of Southern California  
Los Angeles, CA 90089

A searcher and target move among a finite set of cells  $C = 1, 2, \dots, N$  in discrete time. At the beginning of each time period, one cell is searched. If the target is in the selected cell  $i$ , it is detected with probability  $q_i$ . If the target is not in the cell searched, it can not be detected during the current time period. After each search, a target in cell  $j$  moves to cell  $k$  with probability  $p_{jk}$ . The target transition matrix,  $P = [p_{jk}]$  is known to the searcher. The searcher's path is constrained in that if the searcher is currently in cell  $i$ , the next search cell must be selected from a set of neighboring cells  $C_i$ . (The set  $C_i$  is also those cells from which  $i$  can be reached.) The object of the search is to minimize the probability of not detecting the target in  $T$  searches.

## 1. Background

The path constrained search problem, described above, is a difficult one to solve efficiently. Trummel and Weisinger [1986] showed that the path constrained search problem with a stationary target is NP-complete. The moving target problem, which is a generalization of the stationary target problem, is easily shown to be at least as difficult.

The only optimal solution technique mentioned in the literature for this problem has been the dynamic programming procedure of Eagle [1984a]. Although this method can solve problems much more quickly than can total enumeration, it can require a large amount of computer storage as problem size increases.

There have been several approximate solution procedures suggested for this problem. The first such method proposed was a branch-and-bound method by Stewart [1979] and [1980]. Stewart's bounds were obtained by solving an integer problem without path constraints. The solution to this simpler problem was obtained using a discrete version of the moving target search algorithm given by Brown [1980]. However, Brown's algorithm does not necessarily give optimal solutions when search effort is discrete, so these "bounds" are only approximate and can result in an optimal branch of the enumeration tree being mistakenly fathomed. Nonetheless, Stewart's computational experience with 1-dimensional search problems indicates that the method can perform well.

Another approximate procedure was given by Eagle [1984b]. This dynamic programming method uses a moving or "rolling" time horizon that greatly reduces computer storage requirements. It was used to approximately solve a small 2-dimensional problem (3 by 3 search grid) for 40 time periods. This procedure generalizes myopic search by selecting the next cell to be searched under the assumption that the search ends  $m$  time periods in the future. For myopic search,  $m$  is 1. For small enough  $m$ , this procedure can be implemented on a microcomputer.

Reported here is an optimal solution method which appears to perform more efficiently than the existing optimal dynamic programming procedures. This work is a direct continuation of that of Stewart [1979]. The approximate bound in Stewart's branch-and-bound procedure is replaced with a true lower bound, thus guaranteeing optimal solutions. The true bound is obtained by relaxing the integer problem in a manner first suggested by Stewart [1979]. The branch-and-bound procedure is made computationally feasible by special structure in the relaxed problem which allows for its efficient solution.

## 2. The Relaxed Problem

The procedure presented here depends critically on the efficient solution of a convex nonlinear program, here called the *relaxed problem* or RP. The relaxed problem, introduced by Stewart [1979], allows infinite divisibility of search effort over a subset of the search cells, while choosing this subset properly to maintain a representation of path constraints. Following Stewart, let  $x(i, j, t)$  be the amount of search effort that is redistributed from cell  $i$  in time period  $t$  to cell  $j$  in time period  $t + 1$ . And let  $X(i, t)$  be the total search effort in cell  $i$  at time  $t$ . Then,

$$X(i, t) = \sum_{k \in C_i} x(k, i, t - 1), \quad t = 1, 2, \dots, T \quad \text{and} \quad i = 1, \dots, N.$$

The searcher's initial search effort distribution is  $X(i, 0)$ , and is assumed to be specified. For a single searcher (our assumption here), we additionally require  $X(i, 0) = 1$  if  $i$  is the searcher's starting cell, and  $X(i, 0) = 0$  otherwise.

An exponential detection function is assumed for RP. That is, if the search effort in cell  $i$  at time  $t$  is  $X(i, t)$ , then the probability of detecting the target during that search is  $1 - \exp(-\alpha_i X(i, t))$ . The term  $\alpha_i$  is selected so that  $1 - \exp(-\alpha_i) = q_i$ . This insures that integer solutions for RP have the correct objective function values. To keep  $\alpha_i$  finite, we require  $q_i < 1$ . Additionally we let  $\omega = (\omega(1), \omega(2), \dots, \omega(T))$  be a sample target track, and  $p_\omega$  be the probability that the target follows that path. Finally, the set of all possible target paths is  $\Omega$ .

The optimal search flows  $x(i, j, t)$  for RP are obtained by solving the following nonlinear program:

$$\begin{aligned} &\text{minimize} && \sum_{\omega \in \Omega} p(\omega) \exp \left\{ - \sum_{t=1}^T \alpha_{\omega(t)} \sum_{i \in C_{\omega(t)}} x(i, \omega(t), t - 1) \right\} \\ &\text{subject to:} && \\ &&& X(i, 0) - \sum_{j \in C_i} x(i, j, 0) = 0, \quad i = 1, \dots, N \\ &&& \sum_{i \in C_j} x(i, j, t - 1) - \sum_{k \in C_j} x(j, k, t) = 0, \quad j = 1, \dots, N \quad \text{and} \quad t = 1, \dots, T - 1 \\ &&& x(i, j, t) \geq 0, \quad i, j = 1, \dots, N \quad \text{and} \quad t = 0, \dots, T - 1 \end{aligned}$$

As Stewart observed, the objective function for RP is convex, and the constraints are those of a network. The *integer problem* (IP) is identical to RP except that each  $x(i, j, t)$  must be either 1 or 0.

When the target motion is Markovian with transition matrix  $P = [p_{ij}]$ , the objective function of RP can be written

$$\text{minimize} \quad \pi(1) P_1 P_2 \dots P_T \mathbf{1}, \quad (1)$$

where  $\pi(1)$  is a row vector of the target's initial distribution over the search cells  $C$  (assumed to be known),  $\mathbf{1}$  is a column vector of ones, and  $P_t$  is the target transition matrix  $P$  with each row  $i$  multiplied by  $\exp(-\alpha_i X(i, t))$ . The matrix  $P_t$  is substochastic and updates the (defective) target distribution for the search in time period  $t$  and the target transition into time period  $t + 1$ .



As Brown [1980] pointed out, the objective function (1) can be written by conditioning on the target's cell in time period  $t$ . Specifically,

$$\sum_{j=1}^N r(j, t) \exp \left\{ - \sum_{i \in C_j} \alpha_j x(i, j, t-1) \right\} s(j, t), \quad (2)$$

where  $r(j, t)$  is the probability of the target reaching cell  $j$  in time period  $t$  without being detected by searches in time periods  $1, 2, \dots, t-1$ ; and  $s(j, t)$  is the probability of a target which starts in cell  $j$  in time period  $t$  surviving the searches in time periods  $t+1, \dots, T$ . Both  $r(\cdot, t)$  and  $s(\cdot, t)$  are easily calculated from  $r(\cdot, t-1)$  and  $s(\cdot, t+1)$ , respectively. The recursions begin with  $r(i, 1) = \pi_i(1)$  and  $s(i, T) = 1$ , for all cells  $i$ .

The importance of expressing the objective function as in (2) is that it allows a convenient calculation of the gradient. If  $f(\mathbf{x})$  is the objective function, then elements of the gradient  $\nabla f(\mathbf{x})$  are

$$\frac{\partial f(\mathbf{x})}{\partial x(i, j, t)} = -\alpha_j r(j, t+1) \exp \left\{ - \sum_{k \in C_j} \alpha_j x(k, j, t) \right\} s(j, t+1).$$

Note that all gradient elements are nonpositive, since increasing the search flow along any arc can not increase the probability of missing the target.

### 3. Solving the Relaxed Problem

Stewart solves RP essentially as suggested by Frank and Wolfe [1956]. That is, starting with a set of feasible search flows  $\mathbf{x}_1$ , the gradient  $\nabla f(\mathbf{x}_1)$  is computed. Then  $\mathbf{x}'_1$  is found by minimizing the linear constraint  $\nabla f(\mathbf{x}_1)\mathbf{x}$  subject to the network constraints of RP. An improved, feasible set of flows  $\mathbf{x}_2$  is then found by searching along the line segment  $[\mathbf{x}_1, \mathbf{x}'_1]$ . The process continues, generating an improving sequence of flows  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots$  until a stopping criterion is met.

Stewart's solution procedure for RP is particularly effective for the search problem presented here for two reasons:

1. Since this is a single-searcher problem, arc capacity constraints (allowed by Stewart) are not necessary. Also a single initial search cell is specified, rather than a search effort distribution over several cells. Consequently, the linear subproblem reduces to an acyclic, shortest path problem—perhaps the simplest of all nontrivial linear programs to solve. The solution can be obtained by dynamic programming in polynomial time (each arc is examined only once) and does not require a general network solver. We note that the network is acyclic because each arc connects a time  $t$  node to a time  $t+1$  node.

2. At each iteration, the Frank-Wolfe method produces a lower bound for the optimal RP objective function value. In particular, if  $\mathbf{x}_n$  is the current feasible solution to RP, and  $\mathbf{x}^*$  is the optimal solution, then by the convexity of  $f(\mathbf{x})$ ,

$$f(\mathbf{x}^*) \geq f(\mathbf{x}_n) + \nabla f(\mathbf{x}_n)(\mathbf{x}^* - \mathbf{x}_n)^t.$$

And since  $\mathbf{x}'_n$  minimizes  $\nabla f(\mathbf{x}_n)\mathbf{x}$  over all feasible flows  $\mathbf{x}$ ,

$$f(\mathbf{x}^*) \geq f(\mathbf{x}_n) + \nabla f(\mathbf{x}_n)(\mathbf{x}'_n - \mathbf{x}_n)^t.$$

For convex programs with bounded feasible regions (as we have here), Frank and Wolfe [1956] showed that this lower bound for  $f(\mathbf{x}^*)$  becomes exact as  $\mathbf{x}_n$  approaches  $\mathbf{x}^*$ . The bound is important since a lower bound to IP (which is provided by a lower bound to RP) is required for the branch-and-bound solution of IP.

#### 4. Solving the Integer Problem

Stewart [1979] and [1980] gave a branch-and-bound algorithm for IP. The only change suggested here is the use of RP for fathoming. There are, however, at least two important implementation issues. First, when using Stewart's procedure to solve a particular instance of RP, the Frank-Wolfe steps need continue only until branching or fathoming criteria are met. In particular, let PBEST be the branch-and-bound procedure's current best (i.e., smallest) objective function value for an integer solution. Then during the solution of RP,

1. If the lower bound to RP  $(f(\mathbf{x}_n) + \nabla f(\mathbf{x}_n)(\mathbf{x}'_n - \mathbf{x}_n)^t)$  becomes greater than PBEST, the current branch is fathomed.
2. If the upper bound to RP  $(f(\mathbf{x}_n))$  becomes smaller than PBEST, then fathoming is impossible, and the procedure must branch one time period deeper into the enumeration tree.
3. If the difference between the current RP upper and lower bounds  $(\nabla f(\mathbf{x}_n)(\mathbf{x}'_n - \mathbf{x}_n)^t)$  becomes less than some specified  $\epsilon > 0$ , then the procedure branches.

A second observation is that each  $\mathbf{x}'_n$ , which is an optimal solution to a linear subproblem, is an extreme point of the feasible region for RP. Each  $\mathbf{x}'_n$  is thus integer-valued and a feasible solution to IP. These solutions can be used to obtain an initial, integer starting solution and to continually update PBEST in the branch-and-bound procedure.

#### 6. Computational Results

In Eagle [1984a], a 9-cell, 10-time period problem was reported solved to optimality using dynamic programming, and requiring 19.3 CPU minutes on a mainframe computer. This same problem required 2.5 CPU minutes using the branch-and-bound procedure presented here. (The comparison is not entirely fair since the dynamic programming solution was, essentially, for all possible initial target distributions. Once an optimal solution was obtained for any initial target distribution, only an additional 1.5 CPU seconds were required to obtain an optimal solution for any other initial target distribution.) A few larger problems have been attempted. A 25-cell, 10-time period problem required 3.15 CPU minutes; and a 49-cell, 10-time period problem required 4.7 CPU minutes. All computations were done in Fortran 77 on the Naval Postgraduate School IBM 3033 mainframe computer.

## References

- BROWN, S. S. 1980. Optimal Search for a Moving Target in Discrete Time and Space. *Opns. Res.* **28**, 1275-1289.
- EAGLE, J. N. 1984a. The Optimal Search for a Moving Target When the Search Path is Constrained. *Opns. Res.* **32**, 1107-1115.
- EAGLE, J. N. 1984b. The Approximate Solution of a Simple Constrained Search Path Moving Target Problem Using Moving Horizon Policies. Naval Postgraduate School Technical Report NPS55-84-009.
- FRANK, M. and WOLFE, P. 1956. An Algorithm for Quadratic Programming. *Naval Research Logistic Quarterly*, Vol. 3, Nos. 1 and 2.
- STEWART, T. J. 1979. Search for a Moving Target When Searcher Motion is Restricted. *Comput. & Ops. Res.* **6**, 129-140.
- STEWART, T. J. 1980. Experience with a Branch-and-Bound Algorithm for Constrained Searcher Motion. In *Search Theory and Applications*, K. B. Haley and L. D. Stone (eds.). Plenum Press, New York.
- TRUMMEL, K. E. and WEISINGER, J. R. 1986. The Complexity of the Optimal Searcher Path Problem. *Opns. Res.* **34**, 324-327.

# DISTRIBUTION LIST

	<u>NO. OF COPIES</u>
Library (Code 0142) Naval Postgraduate School Monterey, CA 93943-5000	2
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Office of Research Administration (Code 012) Naval Postgraduate School Monterey, CA 93943-5000	1
Center for Naval Analyses 4401 Ford Ave. Alexandria, VA 22302-0268	1
Library (Code 55) Naval Postgraduate School Monterey, CA 93943-5000	1
Operations Research Center, Rm E40-164 Massachusetts Institute of Technology Attn: R. C. Larson and J. F. Shapiro Cambridge, MA 02139	1
Koh Peng Kong OA Branch, DSO Ministry of Defense Blk 29 Middlesex Road SINGAPORE 1024	1
Arthur P. Hurter, Jr. Professor and Chairman Dept of Industrial Engineering and Management Sciences Northwestern University Evanston, IL 60201-9990	1
Institute for Defense Analysis 1800 North Beauregard Alexandria, VA 22311	1
Chief of Naval Operations Washington, DC 20350-2000	1



DUDLEY KNOX LIBRARY



3 2768 00302441 5